Travaux pratiques n° 5 : Contrôle de trafic dans les routeurs

09/04/2013

Ce TP a pour objectif de montrer le rôle des routeurs dans la performance de TCP. Nous allons voir en premier les problèmes occasionnés à TCP par une gestion de file d'attente inappropriée. Ensuite nous verrons les apports d'une gestion active des paquets par les routeurs.

Exercice 1 — Défauts de Droptail

Comment interagissent le réseau et les flots TCP ? Quel est le rôle de la discipline d'attente (*queue management*) dans la performance de TCP ? Nous allons mettre en évidence les principaux défauts de la discipline d'attente dite "Droptail". La figure 1 représente le modèle de réseau de cette étude. Ses caractéristiques sont :
– le tampon mémoire pour le lien 01 est de 22 paquets.
– la fenêtre de contrôle de flux des connexions est limitée à 60 paquets.
– la version de TCP est TCP/Newreno.
– la durée de la simulation est de 50 secondes.
– les paquets ont un taille de 1000 octets.
Vous pourrez utiliser le fichier `ns-tme.tcl` qui contient la configuration du scénario de simulation pour ns-2 [1]. Il est possible de modifier certains paramètres de simulation en les passant en arguments *e.g.* le type de file d'attente, le délai de propagation sur le lien $(n_2 - n_0)$, le paramètre d'overhead (...). Pour lancer une simulation utilisez la commande `ns ns-tme.tcl arguments`. Lisez attentivement le contenu du script `ns-tme.tcl` afin de déterminer les arguments qu'il vous semble nécessaire de spécifier. Lors de chaque simulation, les résultats sont stockés dans un fichier de trace [2] nommé `out.tr` qui conserve les principaux évènements *e.g.* réception d'un paquet sur un lien, mise en file d'attente, réception par le destinataire (...). Afin de faciliter l'analyse du fichier `out.tr`, le script `getStatsFromFlows.sh` calcule la quantité de données envoyées, le débit moyen écoulé et le taux de perte pour chaque flot et pour l'agrégat. Lisez le script afin de déterminer comment sont formatées les données qu'il affiche à l'écran. Le lancement des simulations et le stockage des résultats peuvent être automatisés via le script `launchMultipleExp.sh`. Pour afficher les résultats sous forme de courbes et graphiques, vous pourrez utiliser l'outil `gnuplot` [3].

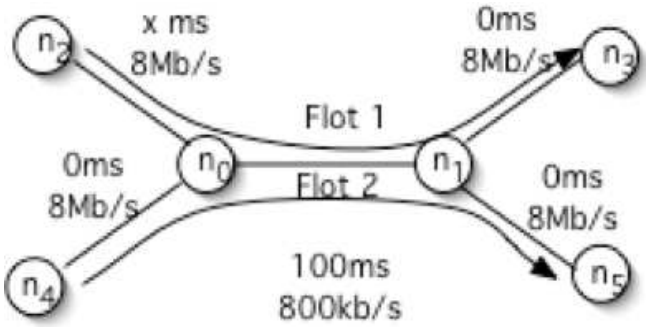

FIGURE 1 – Modèle de réseau à 2 sources.

a. Le lien d'accès pour le flot 1 a un temps de propagation Tp_1 tel que $RTT_1 = \alpha.RTT_2$ avec α un coefficient multiplicateur. Exprimez Tp_1 en fonction de RTT_2 et α. La valeur de RTT_i est déterminée sans la file d'attente. Tracez sur un graphe la courbe du débit écoulé (throughput) et du taux de pertes (dropratio) de chaque flot en fonction de α avec α variant de 1 à 1.1 avec un pas de 0.02. Le partage de la bande passante est il équitable entre les deux flots ? Quelle est la raison de la différence de débit ?

b. Retenez un cas où le partage est le plus inéquitable. Introduire un temps aléatoire de traitement des acquittements (paramètre `overhead`) de valeur maximum de 0.1 seconde. Relancez l'exécution de la simulation. Que remarquez vous ? Pourquoi ? A la lumière de cette dernière expérience, expliquez le phénomène que vous avez constaté à la question 1.

c. Reprenons le modèle de la figure 1 avec $Tp_1 = 0$ et en indiquant un démarrage du flot 2 avec un retard de 5 secondes sur le flot 1. Tracez la fenêtre de congestion de TCP (`cwnd`) de chaque source en fonction du temps. Pour ce faire vous pourrez utiliser le script `printQueueSizeCwndAndTp.sh` qui traite le fichier `cwndAndQueue.tr`. Ce fichier est généré par `ns` via l'observation périodique des valeurs de la cwnd des flots 1 et 2 ainsi que la taille de la file d'attente du lien $(n_0 - n_1)$. Veillez à mettre un paramètre `overhead` dans les sources TCP. Toujours à l'aide du script `printQueueSizeCwndAndTp.sh`, tracez l'évolution de la longueur de la file d'attente en fonction du temps. A l'aide de ces deux graphes, que remarquez vous ? Le goulot d'étranglement est il toujours en train d'écouler du trafic ? Quelle est la taille moyenne de la file d'attente ? La file d'attente déborde-t-elle ?

Pour conclure, à partir de vos observations, indiquez les effets indésirables du DropTail ?

Exercice 2 — Discipline d'attente

Des nouvelles disciplines d'attente ont été proposées afin de remplacer le Droptail. Quels sont leurs apports ? Cette étude vise à mieux comprendre le rôle de la discipline d'attente dans le service best-effort. Pour évaluer les améliorations, nous allons retenir le modèle de simulation représenté par la figure 1 avec $Tp_1 = 0$. Les améliorations seront mesurées par le débit écoulé moyen et l'équité entre les flots. L'analyse de la longueur de la file d'attente pourra servir à déduire le temps d'attente moyen et la variation du temps d'attente

a. Une première idée a été de déporter les pertes de l'arrière de la file à l'avant de la file. Changez la politique de perte de Droptail de la file d'attente du goulot d'étranglement de manière que les pertes s'effectuent par l'avant. La politique de perte s'indique par le booléen drop_front. Pour se faire, ajoutez dans votre fichier des ressources dans la section des valeurs par défauts : `Queue/DropTail set drop_front_ true` Veillez à mettre un paramètre overhead dans les sources TCP. La seconde source démarre avec un retard de 5 secondes. Quel est le gain d'une gestion déterministe de la file avec les pertes par l'avant de la file d'attente ? Au niveau de la longueur de la file d'attente est ce que cela change quelque chose ?

b. L'IETF recommande le déploiement de la discipline d'attente RED (*Random Early Detection*) dans les routeurs de l'Internet [RFC2309]. RED vise à corriger les défauts de Droptail pour aider le contrôle de congestion de TCP. RED repose sur l'idée qu'il ne faut pas attendre que la file d'attente d'une interface de sortie d'un routeur soit pleine pour que les sources détectent la congestion et commencent à agir. Il faut anticiper l'apparition de la congestion lorsque la charge est forte afin d'éviter le débordement de la file d'attente et de jeter les paquets par blocs. RED déduit le niveau de congestion par la taille moyenne de la file d'attente (avg) et vérifie qu'elle se trouve entre un seuil minimum (min_{th}) et un seuil maximum (max_{th}). Si c'est le cas, un paquet qui arrive est jeté avec la probabilité $p = f(avg)$, qui est une fonction croissante de la taille moyenne de la file d'attente. Tous les paquets qui arrivent quand $avg > max_{th}$ sont jetés. La taille moyenne de la file d'attente est calculée selon une moyenne exponentielle (EWMA). Le paramètre avg est initialisé à 0 puis à chaque arrivée de paquet il est actualisé comme suit :

$$avg = (1 - w_q) * avg + w_q * q$$

où q est la taille réelle de la file d'attente et w_q est la pondération. La première version de RED souffrait d'un gros défaut qui portait sur la configuration de ses paramètres. La proposition *Adaptive RED* a corrigé ce défaut. L'auto-configuration est activée par le paramètre `gentle` (quand celui-ci est à la valeur vraie). Par la suite nous retiendrons ce paramètre pour les simulations avec RED. Les paramètres de configuration de RED s'indiquent de la manière suivante :

```
Queue/RED set setbit_ false
Queue/RED set gentle_ true
Queue/RED set mean_pktsize 1000
Queue/RED idle_pktsize 200
```

Soit le même modèle que précédemment sauf que la discipline d'attente au niveau du goulot d'étranglement est maintenant RED. Comparez les performances induites par RED par rapport à Droptail en terme de qualité du service de communication et en terme d'efficacité du réseau :

1. Le manuel d'utilisation est inclus dans l'archive qui accompagne ce TME et il est également disponible en ligne : `http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf`
2. Les format de trace est disponible à l'adresse : `http://nsnam.isi.edu/nsnam/index.php/NS-2_Trace_Formats`.
3. Le manuel de gnuplot est inclus dans l'archive qui accompagne ce TME. Il est également disponible à l'adresse `http://www.gnuplot.info/docs_4.6/gnuplot.pdf`

- Au niveau des flots TCP, quel est le débit écoulé de chaque flot ?
- Comment s'effectue le partage de la bande passante du goulot d'étranglement ? Quel est l'impact de RED sur l'équité entre les deux sources TCP ? Quel est le débit écoulé soutenable par le réseau ? Comment évolue le taux de pertes des paquets ?
- RED a t-il une influence sur la longueur optimale de la file d'attente du routeur ? Tracez l'évolution de la file d'attente du goulot d'étranglement et tracez l'évolution de la cwnd pour chaque flot. Que remarquez vous ?

Enlevez le paramètre overhead des sources TCP et reprenez le cas le pire trouvé à la question 1 de l'exercice précédent. Quelle est l'influence de RED sur ce cas ? Faites une synthèse des améliorations et des raisons de celles-ci que RED apporte au réseau en général.